Programming Assignment 1: Triangle Meshes and Transformations
Computer Graphics, Fall 2016

Submission deadline: Thursday, October 6th, 11:59

In this assignment you will gather practical experience with 3D objects consisting of triangle meshes, and 3D transformations. In addition to the assignment description, we provide Java code on github to facilitate the introduction to 3D programming. Your solution must be based on this code, as it will form the base for further assignments. You have to submit your solution on ILIAS before Thursday, October 6th, 11:59. Grading will take place in a meeting with a teaching assistant, for which you need to register in the inscription list on ILIAS.

Getting Started with the Java Code

If you work on your own computer, you will need the Eclipse IDE. Install the most recent version "Eclipse IDE for Java EE Developers, Neon Release"[1]. If you work on the computers in the ExWi pool, you will need a permanent Home directory for the pool computers. To set this up, send your Campus Account username to Marco Manzi (`mailto:manzi@iam.unibe.ch`).

In order to run the Java code, first create an empty folder in which your project will be located. Start eclipse and select the created folder as workspace. Now you can import the Java code to the workspace by selecting the menu entry "File → Import → Git → Projects from Git → Next". Then choose "Clone URI". In the next dialogue "Source Git Repository", use `https://github.com/mzwicker/Computergrafik-Basecode.git` as URI. Leave the fields "User" and "Password" empty. In the dialogue "Local Destination", choose the created folder. This folder must be empty. After completion of the import you have to wait some seconds for the finalization of the project creation. If the Maven build is not done correctly by default, you have to update the projects by right-clicking on the project name, then "Maven → Update Project...". Afterwards, you should be able to run the project "simple" and see a rotating cube. If you run into any issues, refer to the forum on ILIAS.

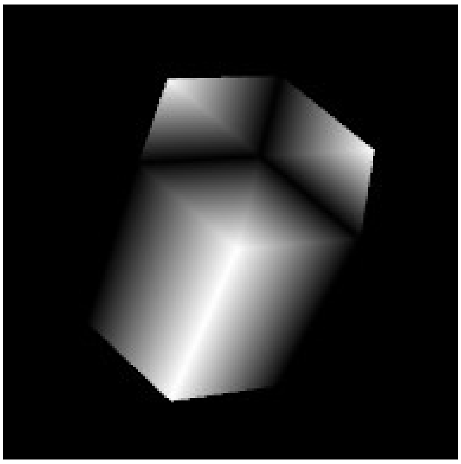
[1] If you use an older version of Eclipse, you may have to install the plugins for "Maven" and "Git" separately. If you run into any issues, refer to the forum on ILIAS.

Get familiar with the structure of the code using the JavaDoc documentation in the folder “doc”, located in the created folder. The folders “obj” and “textures” contain 3D objects and textures, which we will be using later.

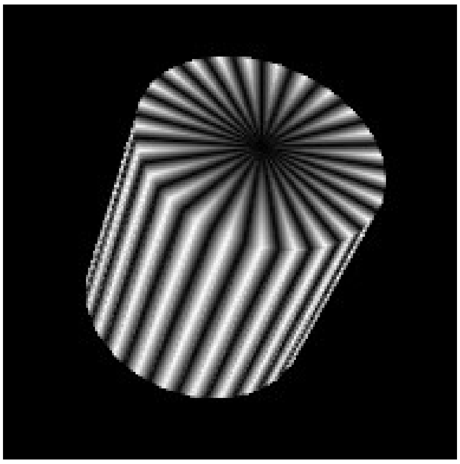
1 Cylinder (3 Points)

Write a function which generates a triangle mesh for a cylinder. The function should return arrays of vertices, colors and indices, which then can be used similarly to the cube in the existing Java code. Your function should accept a parameter for the resolution of the triangles mesh. This parameter indicates how many segments are used to construct the cylinder. The top and the bottom of the cylinder should be closed using a disk. Use a color scheme which assigns different colors to the triangles (see Figures).

Caution: Take care of the order of the vertices for the triangles. The vertices must be ordered counterclockwise, if the surface is observed from the outside. This orientation is defined as “front facing”. If the vertices are ordered clockwise, the triangle is not drawn when being observed from outside. This orientation is called “back facing”.



Cylinder with 6 Segments



Cylinder with 50 Segments

2 Torus (3 Points)

Write a function which generates a torus, similar to the cylinder of exercise 1. The Wikipedia page contains useful information about the mathematical definition of a torus.

3 Animation (4 Points)

Construct an animated object consisting of at least three parts. Use cylinders, tori and cubes. In your animation, at least three parts should be moving relatively to each other. Possible examples are:

- A vehicle with spinning wheels, which drives over a plane in a circle. The wheels move relatively to the vehicle, which in turn moves relatively to the ground plane.
- An airplane with a spinning propeller, which circles above a ground plane.
- A helicopter with spinning rotors, which circles above a ground plane.

Model the scene by transforming (scaling, rotating, translating) the base objects (cylinders, tori, cubes).